

**TITLE OF THE INVENTION**

**COMPUTER SYSTEM AND METHOD CAPABLE OF AUTOMATICALLY  
INPUTTING PRODUCT KEY OF A SOFTWARE PROGRAM UPON  
REINSTALLING THE PROGRAM THEREON**

**CLAIM OF PRIORITY**

This application makes reference to, incorporates the same herein, and claims all benefits accruing under 35 U.S.C §119 from an application entitled *Computer System And Method Capable Of Automatically Inputting Product Key Of A Software Program Upon Reinstalling The Program Thereon* earlier filed in the Korean Industrial Property Office on 23 November 1999, and there duly assigned Serial No. 99-52232 by that Office.

**FIELD OF THE INVENTION**

The present invention relates to a computer system and, more particularly, to a computer system capable of automatically inputting a product key of a software program upon reinstalling the program in the computer system and method of reinstalling a software program.

## BACKGROUND OF THE INVENTION

Every general-purpose computer has an operating system to provide a user interface between a user and the computer and to run other programs on the computer. Operating system programs provide a software platform on top of which other programs, called application programs, are able to run, and perform basic tasks, such as recognizing an input from a keyboard, sending an output to a display apparatus, keeping track of files and directories on disks, and controlling peripheral devices such as disk drives and printers.

Currently, most PC manufacturers provide a bundle of basic and essential software programs already installed in the computers being offered for sale, for user's convenience. Thus, upon purchasing a new computer, a user has no need to install such programs, and all that is required is for the user to register the installed programs with the programs' manufacturers.

If a user bought a computer operated with a Windows® 98 operating system program, he may first add information to some part of the program to his computer if necessary and compose the program to his environment, such as inscription of his name, keyboard composition, time setting, and so on, to complete the installation of the program.

In such a user program install process, a user must correctly input a product key of the program into product key input window, otherwise the program cannot be completely installed and will not operate until the correct product key is input. In other software programs, which must be completely installed by the user, the product key, also known as the product ID, product certificate information or serial number, must be correctly entered before any part of the program can be installed.

Current software programs, however, have complex product key structures (e.g., the Windows® 98 program has a product key of 25 characters), thus there is a high probability that an error occurs in inputting the product key unless a user carefully checks each character of a product key. Also, since software program product keys are typically stored in a specific area of a hard disk in the hard disk drive, should the user have to reinstall the software programs on his/her computer due to a software failure or due to damage to a portion of the hard disk, etc, then the user must manually input the product keys again for the software being reinstalled. However, if the user cannot find the authentication certificate of a program and has no record of the product key of the program, he cannot reinstall the software. Moreover, if the operating system software had to be reinstalled, or a disk recovery operation needed to be performed, and the user had no record of the product key for the operating system software, the computer would be inoperable until either new software was purchased or, if the software was properly registered, the manufacturer could provide the necessary information to the user.

## SUMMARY OF THE INVENTION

Therefore, an object of the present invention is to provide computer systems and methods capable of improving product key retention reliability.

It is another object of the present invention to provide computer systems and methods capable of conveniently reinstalling software programs that require product key input.

According to an aspect of the present invention, a computer system having an operating system program containing product key information includes a first storage step for storing the

1 operating system program, a second storage step for storing the product key information of the  
2 operating system program when the operating system program is first installed in the first storage  
3 step, a write means for writing the product key information in the second storage step, and input  
4 means for reading out the product key information from the memory means and inputting the read-  
5 out information in an information input window for product authentication of the operating system  
6 program when a reinstalled operating system program is matched with an operating system program  
7 corresponding to the product key information written in the memory means.

8 According to another aspect of the present invention, a computer system has a central  
9 processing unit (CPU), a main memory, a BIOS ROM, and an auxiliary memory for storing  
10 information set by the BIOS ROM, and uses an operating system program containing product key  
11 information. A method of writing the product key information in the auxiliary memory when the  
12 operating system program is installed first comprises the steps of making a user manually input the  
13 product key information corresponding to a procedure of installing the operating system program;  
14 executing the product key information writing program; and writing the manually input product key  
15 information into the auxiliary memory.

16 According to further another aspect of the present invention, a computer system has an  
17 operating system program containing product key information and comprising a central processing  
18 unit (CPU), a main memory, an auxiliary memory for storing the product key information, and at  
19 least one auxiliary memory device. A method of automatically inputting the product key information  
20 manually input by a user when the operating system program is first installed comprises the steps  
21 of reading out the product key information from the auxiliary memory; checking whether the read-

1 out product key information is matched with product key information of an operating system  
2 program that will be reinstalled; and, if matched, automatically inputting the product key information  
3 in a product key information input window of the operating system program.

#### 4 BRIEF DESCRIPTION OF THE DRAWINGS

5 A more complete appreciation of the present invention, and many of the attendant  
6 advantages thereof, will become readily apparent as the same becomes better understood by  
7 reference to the following detailed description when considered in conjunction with the  
8 accompanying drawings in which like reference symbols indicate the same or similar components,  
9 wherein:

10 Fig. 1 is a perspective view showing a computer system according to the present invention;

11 Fig. 2 shows a product key input window displayed on a display screen of a computer system  
12 upon installing a software program that requires product key input;

13 Fig. 3 is a block diagram showing a circuit configuration of an embodiment of a computer  
14 system of Fig. 1;

15 Fig. 4 shows a storage region of the CMOS RAM shown in Fig. 3;

16 Fig. 5 shows a format of a program install flag according to the present invention;

17 Fig. 6 shows a first format of a product key code according to the present invention;

18 Fig. 7 shows a second format of the product key code according to the present invention;

19 Fig. 8 shows a code table for use with the code format of Fig. 7;

20 Fig. 9 shows a third format of the product key code according to the present invention;

Fig. 10 shows a code table for use with the code format of Fig. 9;

Fig. 11 is a flowchart showing a program install process according to the present invention;  
and

Fig. 12 is a flowchart showing a program re-install process according to the present invention.

### DESCRIPTION OF THE PREFERRED EMBODIMENT

Fig. 1 shows a computer system 100 according to the present invention, which comprises a system unit 90, a display device 110, an alphanumeric input device 120 such as a keyboard, and a point device 130 such as a mouse. The system unit 90 is equipped with one or more hard disk drives (HDDs) 140 (and 144 of Fig. 3), a compact disk read only memory (CD-ROM) drive 150, a floppy disk drive (FDD) 170, and other mass storage devices such as a digital versatile disk (DVD) player (not shown).

Also, the computer system 100 contains a bundle of software programs including an essential part of an operating system (OS) program and other application programs, all of which are installed by a computer manufacturer or user and stored in the hard disk drive 140.

A commercial software program, such as a Windows® OS program, is usually provided with a manual 10 including an authentication certificate for the program. For example, the authentication certificate of a Windows® 98 OS program contains a product key 12 printed thereon. Additionally, the product key may be printed on a label (not shown) attached to the case 161 for compact disk 160 having the Windows® 98 OS program stored thereon. A product key 12 of a

1 Windows® 98 OS program is composed of 25 characters.

2 Fig. 2 shows a product key input screen 20 displayed on a display screen of display device  
3 110 upon installing a software program that requires the user to input the product key. In such an  
4 installation process, it is required to correctly input a product key 12 of the program in a product key  
5 input window 22 of the product key input screen 20.

6 Fig. 3 shows a circuit configuration of the computer system 100 as shown in Fig. 1.  
7 Referring to Fig. 3, the computer system 100 includes a central processing unit (CPU) 102, a main  
8 (RAM) memory 104, a basic input output system (BIOS) ROM 106, a complementary metal oxide  
9 semiconductor (CMOS) random access memory (RAM) 108 for BIOS setup, a video controller 112,  
10 an input/output (I/O) controller 114, an intelligent or integrated drive electronics (IDE) controller  
11 116, and an FDD controller 118, all of which are coupled to a bus such as a processor bus, a  
12 peripheral component interconnect (PCI) bus, an industrial standard architecture (ISA) bus, or a  
13 system management (SM) bus.

14 I/O controller 114 is coupled to the alphanumeric input device 120 such as a keyboard and  
15 to the pointing device 130 such as a mouse. IDE controller 116 is coupled to one or more hard disk  
16 drives (HDDs) 140 and 144. And, FDD controller 118 is coupled to the FDD 170.

17 BIOS ROM 106 stores a BIOS program that controls the booting operation of the computer  
18 system using HDD 140 or CD-ROM 160 in accordance with the information stored in CMOS RAM  
19 108.

20 As is well known, although the CMOS RAM 108 is a volatile memory, it is operated by a  
21 battery and thus acts like a non-volatile memory. That is, the CMOS RAM 108 always retains the

BIOS setup information without loss of the information upon system power-off.

The computer system 100 also includes a product key input program installed by the manufacturer and stored in the HDD 140. The product key input program writes a product key of a software program into the CMOS RAM 108 used for BIOS setup of the computer 100. The product key input program preferably is installed when an OS program is installed for the first time.

The system 100 further includes a system recovery CD-ROM 160 that contains a system recovery program for the recovery of the software programs stored in the hard disk drive 140. When OS and application programs of the computer 100 become defective, the system recovery program restores those defective programs to their original installation states. According to the present invention, the recovery program automatically writes a product key stored in the CMOS RAM 108 into a product key window 22 when a program is reinstalled.

In the system memory map, the CMOS RAM 108 corresponds to I/O ports 70h-73h. The ports 70h and 71h are the IBM® personal computer standard CMOS RAM ports, and the other ports 72h and 73h are additional ports according to the present invention. The port 72h is used to store an offset value of a location of the CMOS RAM 108 that is subjected to a read or write operation. A data item is read from or written into the location indicated by the offset value via the port 73h. That is, the additional ports 72h and 73h serve as additional index and data ports of the CMOS RAM 108, respectively.

For example, assuming that an offset value is stored in a storage location 38h of the CMOS RAM 108, a processor, such as the CPU 102, reads a data item corresponding to the offset value from the CMOS RAM 108 as shown in the following Table 1.

TABLE 1

MOV AL, 38h ; (38h) ? AL
OUT 72h, AL ; (AL) ? 72h
IN AL, 73h ; (73h) ? AL

First, a content of location 38h (i.e., an offset value) is moved into a register AL in the processor. The processor then writes the offset value stored in its register AL into the index port 72h of the CMOS RAM 108 so that a data item stored in a CMOS RAM location corresponding to the offset value is read out and stored in the data port 73h of the CMOS RAM 108. Thereafter, the processor reads the data item from the data port 73h and stores it into its register AL. In this manner, a product key of an software program such as an OS program, stored in the CMOS RAM 108, can be read out from a specific area thereof under the control of the recovery program.

In the computer system 100, HDDs 140 and 144 act, respectively, as a primary device containing at least one OS program for system booting and a secondary device for optional functions. While the primary HDD 140 stores an OS program 142 and application software programs, the secondary HDD 144 stores a copy version 146 of the recovery program 162 contained in the recovery CD-ROM 160. The copied program 162 is used to restore the OS and application programs to their initial installation states.

CD-ROM drive 150 can be used as a boot device, like HDD 140. Also, when a software program fails, the CD-ROM drive 150 can directly recover or reinstall the programs using the CD-ROM 160. At this time, the CD-ROM drive 150 reads a product key of the software program from CMOS RAM 108, and then automatically inputs it into a product key input window 22 produced by the CD-ROM 160 upon reinstallation of the defective software program.

Fig. 4 shows a storage region 180 of the CMOS RAM 108 shown in Fig. 3. Referring to Fig. 4, the storage region 180 of CMOS RAM 108 includes BIOS areas 182 and 194, a checksum area 184, a program install flag area 186, a product key area 188, a recovery CD booting menu area 190, and a recovery mode information area 192.

The BIOS areas 182 (00h-DFh) and 194 (FEh-FFh) are available to a BIOS only and cannot be employed for other use.

The checksum area 184 (E0h) is used to determine whether a product key written into the CMOS RAM 108 is correct by checking whether a result of summing up contents of the product key storage area 188 with contents of the program install flag area 186 by bytes is zero ('0').

The program install flag area 186 (E1h-E2h) is used for indicating which program is being installed. In other words, a product key presently being stored in the CMOS RAM 108 is provided for which program.

The recovery CD booting menu area 190 (FCh) stores information about the boot device that is determined by BIOS CMOS setup, as shown in Table 2.

TABLE 2

Setting Code	Boot Device
1	FDD
2	HDD
3	CD-ROM drive
4	PCMCIA device
5	USB device
6	Embedded Network device

The recovery mode information area 192 (FDh) is used to store information about functions that will be carried out after system booting using the recovery CD-ROM 160, in which the information is stored by bytes. As shown in Table 3, the area 192 determines whether the system 100 will be restored to its initial state or only an OS program will be installed.

TABLE 3

Setting Code	Contents
0	Restore System To Initial State
1	Installing Only an OS Program

The product key input storage area 188 (E3h-FBh) preferably has a 25-byte storage size and

each character of the product key is preferably represented by an 8-bit ASCII code. However, area 188 may have various sizes enough to retain all the product key characters of the programs installed in the system 100, and each character may be represented by a 6-bit code, a 5-bit code, or other codes defined by computer makers in accordance with the character numbers of product keys.

Fig. 5 shows a format of a program install flag 186a according to the present invention. Referring to Fig. 5, the program install flag 186a is an 8-bit code. In Fig. 5, X symbols mean 'don't care' values, i.e., it doesn't matter if each bit of 4-bit codes has any of the hex values 0h-Fh. The lower 4-bit portion, bit0-bit3, of the flag 186a indicates the types of software programs installed such as Windows® OS programs, or other application programs. For example, the lower bit portion of '0h' indicates that no program is installed, '1h' indicates that Windows® 98 is installed, and '2h' indicates that Windows® 98 SE is installed. the upper 4-bit portion, bit4-bit7, of the flag 186a indicates the bit formats of the product keys. For example, the upper bit portion of '0h' indicates an eight-bit format, '1h' indicates a 6-bit format, and '2h' indicates a 5-bit format. For example, when a Windows® 98 SE OS program is installed, each character of the OS program's product key is preferably encoded into 8 bits with lower 4 bits of '2h' and upper 4 bits of '0h'.

Fig. 6 shows an 8-bit format 188a of a product key code according to the present invention. As shown in Fig. 6, characters of a product key of '43...HBB' are encoded into 8-bit ASCII codes, respectively, and then are stored in the product key area 188 (E3h-FBh) of the CMOS RAM 108. For example, the first character B of ASCII code '01000010' is stored at storage location E3h, and the last character 4 of ASCII code '00110100' is stored at location FBh.

Fig. 7 shows a 6-bit format 188b of the product key code, and Fig. 8 shows a code table 200 for use with the code format 188b. Referring to Fig. 7, each character code has 2 higher-order bits TH and 4 lower-order bits FL. For example, the ASCII character 'A' is represented by a 6-bit code '010001' in accordance with the code table of Fig. 8. This format 188b permits representing the product key having a maximum number of 32 ASCII characters as 24 bytes using a compress conversion process. That is, as shown in Fig. 7, an example of 4 characters are encoded into 3 bytes using the code table 200 of Fig. 8. A 6-bit code of a character is encoded by subtracting 30h from the value of a corresponding ASCII code so that the value of the 6-bit code is in a range of 00h-3Fh. For Example, the ASCII character 'Z' has a hex value of 5Ah; subtracting 30h from 5Ah results in a hex value 2Ah; and the binary value of 2Ah is 00101010. Thus the 6-bit code of the ASCII character 'Z' is 101010. In limiting the 6-bit code to a range of 00h-3Fh (binary values 000000~111111) the product key may include the ASCII characters shown in Fig. 8 with the reservation area being available for the ASCII characters (' and a~o).

Fig. 9 shows a 5-bit format 188c of the product key code, and Fig. 10 shows a code table 210 for use with the code format 188c. Referring to Fig. 9, each character code has 1 higher-order bit OH and 4 lower-order bits FL. For example, the ASCII character 'Y' is represented by a 5-bit code '10000' in accordance with the code table of Fig. 10. This format 188c allows 3 characters to be encoded into 2 bytes using the code table 210 of Fig. 10 and the compress conversion process of Fig. 9, so that it can represent a product key having up to 40 ASCII characters.

Fig. 11 is a flowchart showing a program install process according to the present invention. For convenience of explanation, the process of Fig. 11 will be described using a specific example,

1 i.e., a user install process of an OS program, such as one of the Windows® family, however, it  
2 should be understood that the inventive concept of the present invention is applicable to installation  
3 of other software programs.

4 As described earlier in the background section of this specification, most PC makers  
5 provide computers installed with a bundle of basic and essential software programs. Thus, upon  
6 purchasing a new computer with an OS program and upon turning on the computer, S220, a user may  
7 be required to first add some part of the OS program to his computer such as registering user  
8 information S222.

9 Thereafter, the user may have to input a product key of the OS program, S224, which is  
10 stored in a specific area of HDD 140, so that the installation of the OS program continues, S226. At  
11 this time, a processor, such as CPU 102, executes the product key input program stored in the HDD  
12 140, S228 so that the product key is read out from the HDD 140 and is written into the CMOS RAM  
13 108, S230. After completion of writing the product key into the CMOS RAM 108, the key input  
14 program stored in the HDD 140 is automatically uninstalled stored from the HDD 140, since it is no  
15 longer needed, S232. At this time, the installation of the OS program may not be fully complete, so  
16 in step S234, the OS program continues to be installed until finished.

17 Fig. 12 is a flowchart showing a program recovery (or re-install) process according to the  
18 present invention. Referring to Fig. 12, when the OS program stored in the HDD 140 becomes  
19 defective, first, a processor, such as CPU 102, executes the system recovery program 162 stored in  
20 the system recovery CD-ROM 160 by using the CD-ROM drive 150, or system recovery program  
21 146 stored in the HDD 144, in order to restore the defective program to its original installation state,

1 S240. Next, the processor reads out the product key of the defective OS program from the CMOS  
2 RAM 108, S242.

3 After a checksum of the read-out product key is checked, S244, to identify the correctness  
4 of the read-out data, it is determined whether or not the read-out product key is matched with the  
5 product key of the OS program, S246, stored in the system recovery program (162 or 146). If so, the  
6 read-out product key is input to a product key input window 22 for reinstallation of the OS program,  
7 S248, but if not, the processor instructs the user to input the product key of the OS program, S250.

8 The foregoing description of the embodiments of the present invention has been presented  
9 for purposes of illustration and description. It is not intended to limit the present invention to the  
10 embodiments disclosed, and obviously many modifications and variations are possible in light of the  
11 above teachings.